

## Задача Н. Тупики

*Максимальный объем используемой памяти: 64 мегабайта*

*Максимальное время работы на одном тесте: 4 секунды*

На вокзале есть  $K$  тупиков, куда прибывают электрички. Этот вокзал является их конечной станцией, поэтому электрички, прибыв, некоторое время стоят на вокзале, а потом отправляются в новый рейс (в ту сторону, откуда прибыли).

Дано расписание движения электричек, в котором для каждой электрички указано время ее прибытия, а также время отправления в следующий рейс. Электрички в расписании упорядочены по времени прибытия. Поскольку вокзал — конечная станция, то электричка может стоять на нем довольно долго, в частности, электричка, которая прибывает раньше другой, отправляться обратно может значительно позднее.

Тупики пронумерованы числами от 1 до  $K$ . Когда электричка прибывает, ее ставят в свободный тупик с минимальным номером. При этом если электричка из какого-то тупика отправилась в момент времени  $X$ , то электричку, которая прибывает в момент времени  $X$ , в этот тупик ставить нельзя, а электричку, прибывающую в момент  $X+1$  — можно.

Напишите программу, которая по данному расписанию для каждой электрички определит номер тупика, куда прибудет эта электричка.

### **Формат входных данных**

Во входном файле записаны число  $K$  — количество тупиков и число  $N$  — количество электропоездов ( $1 \leq K \leq 500000$ ,  $1 \leq N \leq 500000$ ). Далее идет  $N$  строк, в каждой из которых записано по 2 числа: время прибытия и время отправления электрички. Время задается натуральным числом, не превышающим  $10^9$ . Никакие две электрички не прибывают в одно и то же время. Но при этом несколько электричек могут отправляться в одно и то же время. Также возможно, что какая-нибудь электричка (или даже несколько) отправляются в момент прибытия какой-нибудь другой электрички. Время отправления каждой электрички строго больше времени ее прибытия.

Все электрички упорядочены по времени прибытия. Считается, что в нулевой момент времени все тупики на вокзале свободны.

### **Формат выходных данных**

В выходной файл выведите  $N$  чисел — по одному для каждой электрички: номер тупика, куда прибудет соответствующая электричка. Если тупиков не достаточно, чтобы организовать движение электричек согласно расписанию, в выходной файл должно быть выведено два числа: первое должно равняться 0 (нулю), а второе содержать номер первой из электричек, которая не сможет прибыть на вокзал.

### **Примеры**

h.in	h.out
1 1 2 5	1
1 2 2 5 5 6	0 2
2 3 1 3 2 6 4 5	1 2 1

**Анализ.**

Какие действия нам надо исполнять, какие события у нас происходят?

Во-первых, приезжают электрички. Приезжает электричка – её надо ставить в пустой тупик с наименьшим номером. Ага, строим min-кучу с номерами свободных тупиков. Когда приедет электричка, берём из кучи наименьший номер свободного тупика, и вынимаем корень кучи – это сделать легко, даже локаторы не нужны.

Во-вторых, электричка уезжает, освобождая тупик. Надо знать когда уезжает ближайшая электричка. Ага, из занятых тупиков выбираем тот, который освобождается первым. Про каждый занятый тупик мы знаем когда он освобождается. Значит, собираем вторую кучу из занятых тупиков. Min-кучу, в которой каждый элемент – это пара - номер тупика и время освобождения, причём время – это как раз то, по чему идёт сравнение элементов кучи.

Итого, получаем решение. Берём два момента времени: первый момент – это ближайший момент отъезда электрички, его мы берём из кучи занятых тупиков; второй - это ближайший (минимальный из оставшихся) момент прихода ближайшей электрички, его и искать не надо, электрички уже отсортированы по времени прихода. Какое событие происходит раньше, то и обрабатываем.

Обработка прихода электрички. Берём из кучи свободных тупиков тупик с наименьшим номером, вынимаем его из этой кучи и ставим его вместе со временем отхода этой электрички во вторую кучу – кучу занятых тупиков. Если куча свободных тупиков пустая, то процесс обламывается.

Обработка отхода электрички. Вынимаем корневой тупик из кучи занятых тупиков и ставим его в кучу свободных тупиков.

Вот и всё.